Underleaf

How to Include Code in LaTeX

Including code in your LaTeX documents is essential for technical documentation, programming tutorials, and academic papers. This guide covers all methods from simple inline code to advanced syntax highlighting.

Quick Overview of Methods

METHOD	USE CASE	SYNTAX HIGHLIGHTING
`\verb`	Inline code snippets	No
`verbatim`	Simple code blocks	No
`listings`	Code with basic highlighting	Yes (basic)
`minted`	Professional syntax highlighting	Yes (advanced)

Method 1: Inline Code with \verb

For short inline code snippets, use the `\verb` command. The delimiter can be any character that doesn't appear in your code:

```
Use \verb|print("Hello")| to output text.
The function \verb+array[i]+ accesses element i.
```

Output: Use `print("Hello")` to output text. The function `array[i]` accesses element i.

Method 2: Simple Code Blocks with verbatim

For basic code blocks without any formatting, use the `verbatim` environment:

```
\begin{verbatim}
def hello_world():
    print("Hello, World!")
```

```
hello_world()
\end{verbatim}
```

```
def hello_world():
    print("Hello, World!")

hello_world()
```

Method 3: The listings Package

The `listings` package provides syntax highlighting and many customization options. First, add it to your preamble:

```
\usepackage{listings}
\usepackage{xcolor}

% Define colors for syntax highlighting
\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}

% Configure listings style
\lstdefinestyle{mystyle}{
    backgroundcolor=\color{backcolour},
    commentstyle=\color{codegreen},
    keywordstyle=\color{magenta},
    numberstyle=\tiny\color{codegray},
    stringstyle=\color{codepurple},
    basicstyle=\ttfamily\footnotesize,
    breakatwhitespace=false,
    breaklines=true,
    captionpos=b,
    keepspaces=true,
    numbers=left,
    numbersep=5pt,
    showspaces=false,
    showstringspaces=false,
    showtabs=false,
    tabsize=2
}
```

```
\lstset{style=mystyle}
```

Then use it in your document:

```
\begin{lstlisting}[language=Python, caption=Python example]
def fibonacci(n):
    """Calculate the nth Fibonacci number"""
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)

# Print first 10 Fibonacci numbers
for i in range(10):
    print(fibonacci(i))
\end{lstlisting}
```

Supported Languages in listings

The listings package supports many programming languages:

```
Python   Java   C   C++   JavaScript   HTML   CSS   SQL   PHP   Ruby   Perl   R

Matlab   Bash   TeX
```

Method 4: The minted Package (Advanced)

For professional-quality syntax highlighting, use the `minted` package. It uses Pygments for highlighting, which supports 300+ languages:

The minted package requires Python and Pygments to be installed. You also need to compile with `--shell-escape` flag.

```
% In preamble
\usepackage{minted}

% In document
\begin{minted}[linenos, bgcolor=lightgray]{python}
import numpy as np
```

```
def matrix_multiply(A, B):
    """Multiply two matrices using NumPy"""
    return np.dot(A, B)

# Example usage
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])
print(matrix_multiply(A, B))
\end{minted}
```

Including Code from External Files

Both listings and minted can include code directly from files:

```
% Using listings
\lstinputlisting[language=Python]{code/example.py}

% Using minted
\inputminted{python}{code/example.py}

% Include only specific lines
\lstinputlisting[language=Python, firstline=10, lastline=25]{code/example.py}
\inputminted[firstline=10, lastline=25]{python}{code/example.py}
```

Inline Code with listings

For inline code with highlighting, use `\lstinline`:

```
The \lstinline{for} loop iterates over the \lstinline{range(10)}.
Use \lstinline[language=Python]|print()| to display output.
```

Customizing Code Appearance

Common customization options for listings:

OPTION	DESCRIPTION	EXAMPLE
`numbers`	Line numbers position	`left`, `right`, `none`

OPTION	DESCRIPTION	EXAMPLE
`frame`	Frame around code	`single`, `lines`, `none`
`caption`	Code caption	`caption=My Code`
`label`	Reference label	`label=lst:example`
`breaklines`	Wrap long lines	`true` / `false`

Complete Example Document

```
\documentclass{article}
\usepackage{listings}
\usepackage{xcolor}

\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}

\lstdefinestyle{mystyle}{
    backgroundcolor=\color{backcolour},
    commentstyle=\color{codegreen},
    keywordstyle=\color{blue},
    basicstyle=\ttfamily\small,
    breaklines=true,
    numbers=left,
    frame=single
}
\lstset{style=mystyle}

\begin{document}

\section{Code Example}

Here is a Python function:

\begin{lstlisting}[language=Python, caption=Factorial function]
def factorial(n):
    """Calculate factorial recursively"""
    if n <= 1:
        return 1
    return n * factorial(n - 1)
\end{lstlisting}
```

```
As shown in Listing 1, the \lstinline{factorial} function uses recursion.

\end{document}
```

Best Practices

- Use `verbatim` for quick, simple code snippets without highlighting needs
- Use `listings` for most use cases - it's reliable and doesn't require external dependencies
- Use `minted` when you need the best syntax highlighting and don't mind the setup
- Keep code examples concise and relevant to the document
- Use captions and labels to reference code listings in your text
- Consider using a monospace font that clearly distinguishes similar characters (0/O, 1/l/I)

> When working with Overleaf, the listings package works out of the box. For minted, enable "Shell escape" in the project settings.

> **Tip:** Need help configuring code blocks? Underleaf's AI can generate properly formatted listings and minted environments from your code snippets instantly.

Learn LaTeX

GETTING STARTED

What is LaTeX?

LaTeX for Beginners

TEXT FORMATTING

Bold Text in LaTeX

Italics in LaTeX

Underlining in LaTeX

Subscript and Superscript

Greek Letters and Symbols

MATHEMATICAL EXPRESSIONS

Math Mode Basics

Mathematical Symbols

Matrices and Arrays

Vectors

Vector Spaces

Tensors

Integrals and Summations

Absolute Value

Fractions

DOCUMENT STRUCTURE

Tables in LaTeX

Tables Tutorial

Lists & Enumerations

Code in LaTeX

Style Files (.sty)

Accessibility

Underleaf

Empowering students and researchers with AI-powered tools for academic writing.

Go to app →

Contact us

Company

Pricing

Blog

Tutorials

Affiliate Program

Use Cases

Enterprises

Universities

Research Teams

Accessibility

Free Tools

Image to LaTeX

Excel to LaTeX

ArXiv to LaTeX

TikZ Generator

Thesis Generator

Chrome Extension

All Tools

Support

Support Center

FAQ

Browser Extension

Privacy Policy

© 2026 Underleaf. All rights reserved.